

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-30373

(43)公開日 平成7年(1995)1月31日

(51)Int.Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
H 0 3 H 17/02	P	8842-5 J		
G 0 6 F 7/38	Y			
17/10				
		9364-5 L	G 0 6 F 15/ 31	D

審査請求 未請求 請求項の数1 F D (全 7 頁)

(21)出願番号 特願平5-193113

(22)出願日 平成5年(1993)7月7日

(71)出願人 000004260

日本電装株式会社

愛知県刈谷市昭和町1丁目1番地

(72)発明者 原 光雄

愛知県刈谷市昭和町1丁目1番地 日本電装株式会社内

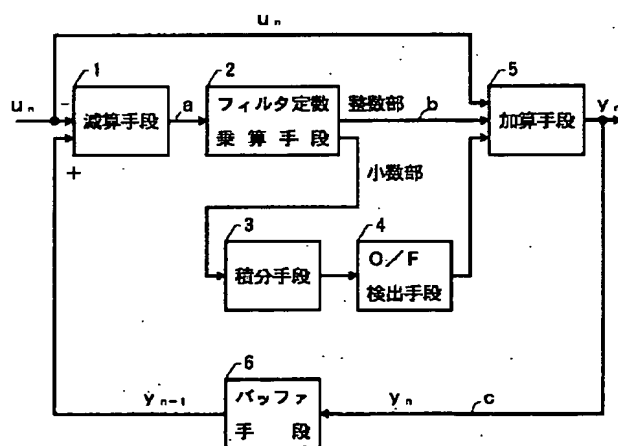
(74)代理人 弁理士 藤谷 修

#### (54)【発明の名称】 デジタルフィルタ

##### (57)【要約】

【目的】デジタルフィルタの桁落ちを補償し、正確な制御追従を実現する。

【構成】図1は一次遅れデジタルフィルタの場合のブロック図で、ローパスフィルタ(LPF)である。一次遅れは、前回の出力値と今回の入力値との差を適切に定数倍した量に、今回の入力値を加えて、今回の出力とする。ここで、定数倍する所で桁落ちが発生するので、乗算手段2で演算されるごとに生じる小数部分を積分手段3で積算累積し、その積算結果が±1以上なら、オーバーフロー検出手段4で±1の出力信号を加算手段5に送る。加算手段5では、通常のデジタル化された整数部の演算に加えて、この±1の出力補正を加算することで、デジタル化による桁落ちを補償する。この補正は偏差が1を越えるごとに発生し、そのたびごとに補正されていくので制御が常に目標値に達することが可能である。



$$a : y_{n-1} - u_n$$

$$b : KFIL * (y_{n-1} - u_n)$$

$$c : u_n + KFIL * (y_{n-1} - u_n) + \text{キャリーフラグ}$$

O/F; オーバーフロー

0/F時 ±1  
0/F時以外 0

## 1

## 【特許請求の範囲】

【請求項 1】乗算要素を有するデジタルフィルタにおいて、

前記乗算要素のフィルタ定数を乗じる乗算手段と、  
前記乗算手段の乗算結果のうち、小数部に相当する下位  
バイトの値を、一回の乗算毎に積分する積分手段と、  
前記積分手段の演算結果で、±1 以上に値がオーバーフ  
ローしたことを検出し、オーバーフローが発生した時の  
み±1 を出力し、それ以外の時には 0 を出力するオーバ  
ーフロー検出手段と、

前記乗算結果のうち、整数部に相当する上位バイトの値  
に、前記オーバーフロー検出手段の出力を加算する加算  
手段を有することを特徴とするデジタルフィルタ。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、コンピュータ自動制御  
等に用いられるデジタルフィルタに関し、また特に、一  
次以上の遅れ特性を有するデジタル式 n 次フィルタに関  
する。

## 【0002】

【従来の技術】従来より、デジタルフィルタとして、図  
7 (a) に示すブロック図のものが提供されている。これ  
は例えば数 1 式で示されるような差分関係を有する。

$$\text{【数 1】 } y_n = y_{n-1} + K \cdot x_n$$

即ち、このフィルタでは、小数部が  $K \cdot x_n$  の項でデジ  
タイズによって桁落ちし、この結果が  $y_{n-1}$  に残って、  
次の演算に影響する。つまり過去の桁落ちの影響を受  
け、誤差が積算されていく問題がある。

【0003】また、図 7 (b) のような構成では（特開平  
2-166913号公報）、これは低域補償用（ハイパス）フ  
ィルタ用の桁落ち防止手段であり、

$$\text{【数 2】 } a_n = a_{n-1} + x_n$$

$$\text{【数 3】 } y_n = K \cdot a_n$$

の数 2 式、数 3 式のように示されるが、この場合でも数  
3 式の項でデジタル化の桁落ちが生じてしまう。ただこ  
の場合では過去の影響を受けないという利点はある。

【0004】しかし、上記の場合に一次遅れのローパス  
フィルタを当てはめてみる。まず一次遅れのローパスフ  
ィルタを式で表すと、数 4 式で示される。

$$\text{【数 4】 } Y(s) = 1 / (1 + TS) \cdot X(s)$$

これを Z 変換して、

$$\text{【数 5】 } y = (1 - K) / (1 - KZ^{-1}) \cdot x$$

が得られる。ここで、K は、

$$\text{【数 6】 } K = \exp(-\Delta t / T)$$

（但し  $\Delta t$  はサンプリング周期）である。従って、図 7  
(c) のようなブロック図となる。この数 5 式は、

## 【数 7】

$$y_n = K \cdot y_{n-1} + (1 - K) \cdot x_n \\ = x_n + K (y_{n-1} - x_n)$$

のような形になって、等価変換しても図 7 (b) の形にな

## 2

らないので、ローパスフィルタには適用できないことが  
わかる。つまり、この提案は一次遅れフィルタ処理には  
用いることができない。

【0005】また、特開平 4-316208 号公報の提案では、  
デジタル一次遅れフィルタの桁落ち誤差を補償する提案  
であるが、フィルタ定数が小さい値、例えば 0.1 であ  
ると、制御は細かい変化に対応できるが、出力誤差の単  
位 1 が強制的に切り替わるため制御対象が最大 10 デジ  
ット分の変化でステップ状の変化を受けることになり

10 （図 8 の h に模式図として示す）、大きな変動を制御対  
象に与えてしまい、図 8 の i に示すような変動を制御対  
象に与えてしまうという問題がある。

## 【0006】

【発明が解決しようとする課題】上記のように、提案さ  
れている方法では、一般的なフィルタに応用することは  
完全とは言えず、特に一次遅れデジタルフィルタにおい  
ては制御性を保ちつつ、誤差を解消するという手段が見  
込めなかったことが課題である。従って本発明の目的  
は、桁落ちを生じる一般的なデジタルフィルタにおい  
て、演算負荷とメモリ効率を悪化させずに、また制御性  
を悪化させることなく、桁落ちによる誤差をなくす補正  
を与えることにある。

## 【0007】

【課題を解決するための手段】上記の課題を解決するた  
めの本発明の構成は、乗算要素を有するデジタルフィル  
タにおいて、前記乗算要素のフィルタ定数を乗じる乗算  
手段と、前記乗算手段の乗算結果のうち、小数部に相当  
する下位バイトの値を、一回の乗算毎に積分する積分手  
段と、前記積分手段の演算結果で、±1 以上に値がオー  
バーフローしたことを検出し、オーバーフローが発生し  
た時のみ±1 を出力し、それ以外の時には 0 を出力する  
オーバーフロー検出手段と、前記乗算結果のうち、整数  
部に相当する上位バイトの値に、前記オーバーフロー検  
出手段の出力を加算する加算手段を有することである。

## 【0008】

【作用】桁落ちして、従来無視していた小数点以下の量  
をサンプルごとに積算して、その量が±1 を越える場合  
は、累積誤差が制御最小単位量を越えることになるの  
で、制御量に±1 の補正をする。即ち乗算要素でデジ  
タイズによる小数点以下の数値が切捨てられるので、その  
演算を無視しないで保存、積算し、その誤差累積が 1 を  
越えない様に補正する。

## 【0009】

【発明の効果】本発明は、乗算要素を含んで桁落ちを生  
じる、どのデジタルフィルタにも応用できる補正であ  
り、とりわけ定常的な制御指示値に対して必ず追従した  
値をとることができ、デジタイズ誤差を残さない。この  
演算のために複雑な変更、メモリ等を必要とせず簡単に  
実現できる。また、補正によって制御異常を生じない。

## 【0010】

## 3

【実施例】以下、本発明を具体的な実施例に基づいて説明する。図 1 は本発明を一次遅れデジタルフィルタに応用した場合のブロック図を示している。このフィルタはローパスフィルタで、例えば図 3 に示すような自動車のスロットルの自動制御に用いられりする。このローパスフィルタのフローチャートの一例を図 4 に示す（後述）。一次遅れは図 7 (c) に示すようなブロック図で示され、数 7 式のような漸化式で表示される。この式の意味は、前回の出力値と今回の入力値との差を適切に定数倍した量に、今回の入力値を加えて、今回の出力とすることである。

【0011】ここで、デジタルフィルタの特性として定数倍するところで桁落ちが発生してしまう。これは、マイクロコンピュータで乗算の演算した結果において、整数部分と小数部分とが得られるが、通常、小数部を無視することから生じる。そこで本発明では、図 1 に示すように、乗算手段 2 で演算されるごとに生じる小数部分を積分手段 3 で積分、累積し、その積分結果の絶対値が 1 を越えたら、オーバーフロー検出手段 4 で  $\pm 1$  の出力信号を加算手段 5 に送る。加算手段 5 では、通常のデジタル化された整数部の演算に加えて、この  $\pm 1$  の出力補正を加算することで、デジタル化による桁落ちを補償する。この補正は偏差が 1 を越えるごとに発生し、そのたびごとに補正されていくので制御が常に目標値に達することが可能である。

【0012】このオーバーフローの発生は、ゲインの値設定によって、つまりシステムによって異なる。これは、以下のように説明される。即ち、 $+1$  デジットが 1 の制御量の変化に対してフィルタ定数（ゲイン）が 0.1 であれば、 $0.1 \times 1 = 0.1$  が積分されていくので、10 回のサンプリングの後にオーバーフローすることになる。フィルタ定数が 0.25 であれば  $+1$  デジットに対する 4 回のサンプリングでオーバーフローとなる。このようにゲインが細かい程サンプリングの回数も増え、ちょうど、漸近線の如くに制御目標に追従していく。

【0013】この一次遅れデジタルフィルタを図 3 に示す車両のスロットルにおいて ISC（アイドルスピードコントロール）に適用する場合について説明する。図 3 で、車両 21 のエンジン 22 にあるスロットル 23 にスロットルアクチュエータ 24 が設けられ、ECU（電子制御回路）26 からの信号でスロットルバルブ 23 を制御してエンジン回転を所定の値に保つ。エンジン回転を指示するアクセルにはアクセルポジションセンサー 25 が設けられ、操作者のアクセルの位置に応じた信号が ECU 26 の入力 I/F 28 に接続されて、エンジン回転数が電気信号として入力される。ECU 26 では ROM 29 に記憶されたプログラムに従って、CPU 30 が、図示しないスロットルセンサー等からの信号を基に、制御すべきスロットルバルブの開度を演算し、駆動回路 2

## 4

7 から駆動信号をアクチュエータ 24 に送って制御する。

【0014】この演算の際に、出力がスロットルバルブの角度というアナログ値であるのに対して、演算はマイクロコンピュータでデジタルで行うため、どうしてもデジタルの桁落ちは避けられない。また、スロットル制御の特性として、わずかな制御量に対して比較的大きなエンジン回転変化となるため、指示量に対してオーバーシュートするような制御ではエンジン回転が不安定になる。そのため、より正確な制御性を求めると共に、ここで一次遅れフィルタを用い、急激な変化に対してある程度ゆるやかな追従をさせ、車両の滑らかな発進、加速を実現させる。その際、この一次遅れデジタルフィルタにおいて桁落ちが累積すると正確な制御が望めないことになるので、本発明が適用される。

【0015】この ISC の基本的な動作をフローチャートにしたものが図 5 に示してある。ステップ 202 で、まずアクセルの位置を検出し、操作者によって決められるエンジン回転指示を電気信号の形  $A_p$  で取り込む。ステップ 204 で目標のスロットル開度  $T_{TA}$  を ECU 内の ROM のマップもしくは数式によって求め、ステップ 206 で目標制御量  $X$  を設置する。そしてステップ 208 で適切なフィルタ定数を設定し、ステップ 210 の LPF（ローパスフィルタ）ルーチンで演算し、ステップ 212 で実際の制御量を出力する。

【0016】LPF ルーチンは一例として図 4 のフローチャートで示される。まずステップ 102 で、入力値  $X$ （図 3 の適用例の場合は目標のスロットル開度  $T_{TA}$ 、図 1 のブロック図では  $u_n$ ）を基に  $dX$  を計算し、つまり数 7 式の後半の  $(y_{n-1} - x_n)$  を求める。そしてステップ 104 でフィルタ定数  $K_{LPF}$  を乗じ、ステップ 106 でその整数部上位 2 バイトと小数部下位 2 バイトを別々に扱い、ステップ 108 で小数部の積分を計算し、この値  $IdY$  がオーバーフローしたかどうかをステップ 110 で判定する。オーバーフロー、即ち誤差が制御量 1 単位を越えたときは、ステップ 114 で制御量に 1 を加算補正する（ $-1$  であれば、 $-1$  を加算、即ち 1 を減算）。なお、このフローチャートの例では、 $IdY$  のレジスタが 1 を越えると、越えた値から 1 を引いた値が自動的に残るレジスタとしているので、 $IdY$  のレジスタをリセットするステップは表示していない。そして、出力値をバッファにセットして前回値として記憶し、LPF の演算が終了する。この図 3 の応用例では、LPF 出力のゲイン補正  $K_G$  を乗ずるステップ 120 を設けている。

【0017】本実施例は、一次遅れのデジタルフィルタに適用したが、一般的なデジタルフィルタの桁落ちを生じる部分に適用することができる。また図 2 に示すように一次遅れ LPF を  $n$  個連ねて、 $n$  次フィルタとしても、それぞれの桁落ちが補償されているので、全体とし

5

ても桁落ち誤差は累積しない。この場合、図4のフローチャートではステップ102からステップ116までが一段のLPFになり、118の流れまでがn回繰り返されることでn次フィルタとなる。これにさらに図4のステップ120のようにゲイン補正をかける場合は、図6のようにn次フィルタの最後に行うことで最終の桁落ちのみ考慮すればよい。このゲイン補正が初段もしくは中段に含まれると、そのゲイン補正部で生じた桁落ちが後段に入り込むため、望ましくない。

【0018】以上のように、本発明はデジタルフィルタの桁落ちを補償し、正確な制御追従を実現させる。この演算のために必要となるのは定数を乗じた際の小数部を積算するメモリのみで、プログラムのステップもわずかに追加されるのみであり、負担がかからない大きな利点がある。従って定数を適切に設定して追従性の優れた制御を実現できる。

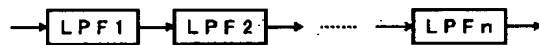
#### 【図面の簡単な説明】

【図1】本発明のデジタルフィルタのブロック図。

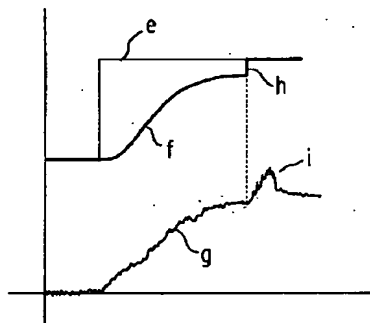
【図2】n次デジタルフィルタの一例のブロック図。

【図3】車両のアイドルスロットル制御（ISC）のブロック図

【図2】



【図8】



e: 目標値 (ステップ応答)

f: 制御値 (一次遅れ)

g: 被制御量

【図6】



6

【図4】デジタル・ローパスフィルタ・ルーチンのフローチャート図。

【図5】図3のISCのフローチャート図。

【図6】n次デジタルフィルタにゲイン補正を加える場合のブロック図。

【図7】従来のデジタルフィルタのブロック図。

【図8】従来提案によりデジタル一次遅れローパスフィルタの桁落ち誤差を補償する場合の動作チャート図。

#### 【符号の説明】

- 1 減算手段
- 2 乗算手段
- 3 積分手段
- 4 オーバーフロー検出手段
- 5 加算手段
- 6 バッファ手段

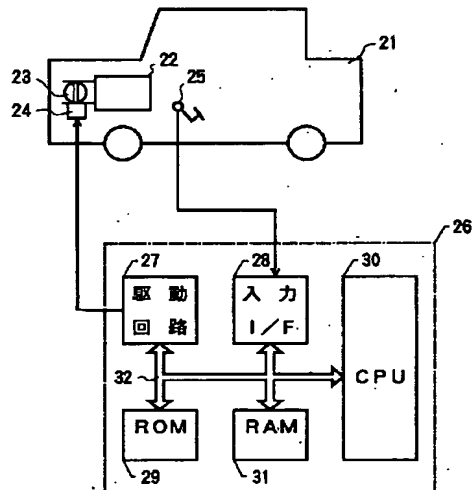
ステップ210 デジタル一次遅れローパスフィルタ・ルーチン

ステップ108 積分手段

ステップ110、112、114 オーバーフロー検出手段

ステップ120 ゲイン補正

【図3】



21: 車両

22: エンジン

23: スロットルバルブ

24: スロットルアクチュエータ

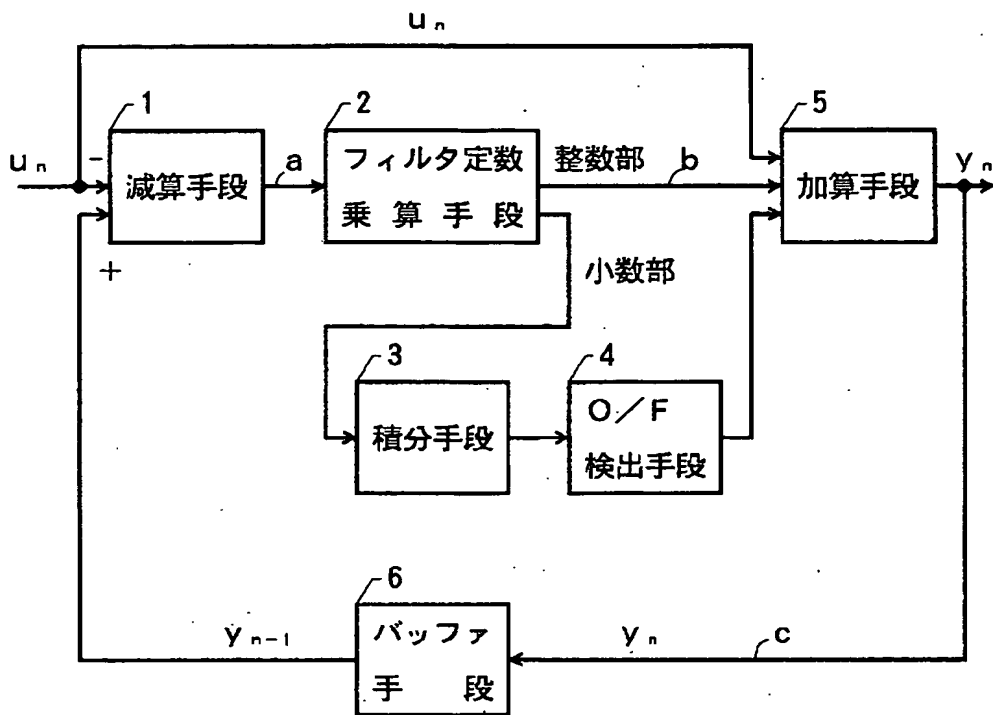
25: アクセルポジションセンサ

26: ECU (電子制御回路)

27: スロットルアクチュエータ駆動回路

32: 内部バス

【図1】



$$a : y_{n-1} - u_n$$

$$b : KFIL * (y_{n-1} - u_n)$$

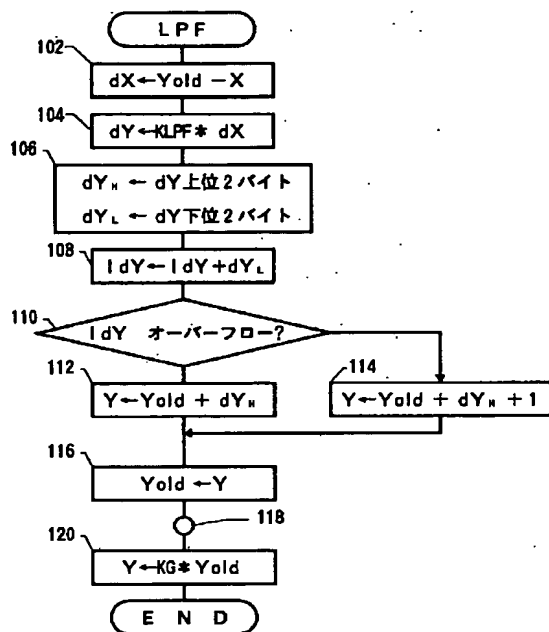
$$c : u_n + KFIL * (y_{n-1} - u_n) + \text{キャリーフラグ}$$

O/F ; オーバーフロー

O/F時  $\pm 1$

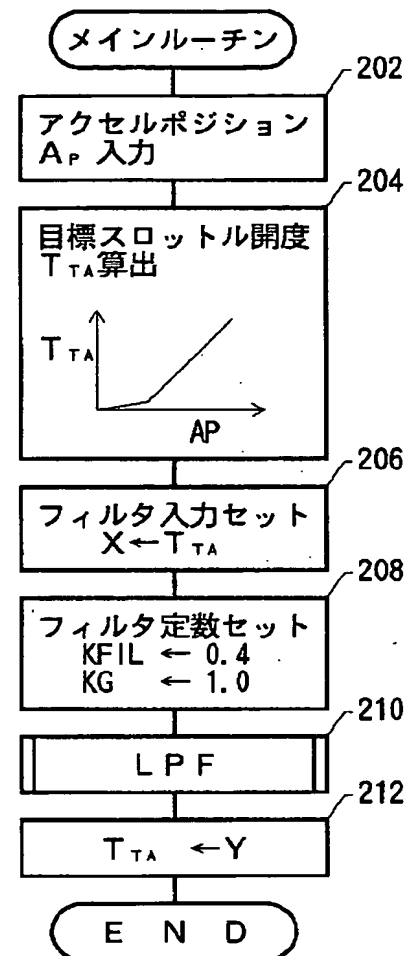
O/F時以外 0

【図 4】



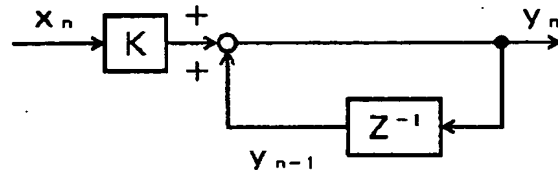
$X$ : 入力データ     $Y$ : 出力データ     $Yold$ :  $Y$  前回値 (2バイト)  
 $dX$ : 入力データの今回値と前回値との差 (4バイト)  
 $dY$ : 出力データの増分 (4バイト)  
 $dY_H$ :  $dY$  上位 2 バイト     $KLPF$ : フィルタ定数 (2バイト)  $LSB=1/2^{10}$   
 $dY_L$ :  $dY$  下位 2 バイト     $KG$ : フィルタゲイン (2バイト)

【図 5】



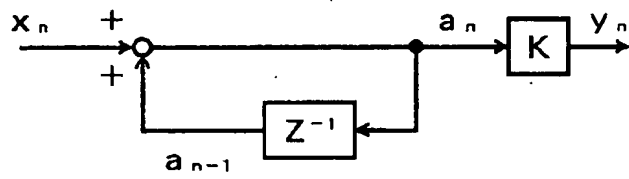
【図7】

(a)



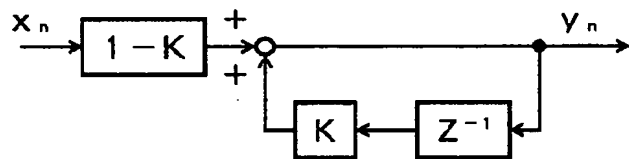
$$y_n = y_{n-1} + K * x_n$$

(b)



$$a_n = a_{n-1} + x_n, \quad y_n = K * a_n$$

(c)



$$\begin{aligned} y_n &= K * y_{n-1} + (1-K) * x_n \\ &= x_n + K * (y_{n-1} - x_n) \end{aligned}$$